Web

Results 1 - 10 of about 102,000 for tim harris pragmatic implementation. (0.34 seconds)

Recent work
Anthony Discolo, **Tim Harris**, Simon Marlow, Simon Peyton-Jones, Satnam Singh. ... A **Pragmatic Implementation** of Non-Blocking Linked Lists October 2001 ...
research.microsoft.com/~t**harris**/ - 26k - Cached - Similar pages

A **Pragmatic Implementation** of Non-Blocking Linked-Lists - **Harris** ...
Timothy L. **Harris**. A **pragmatic implementation** of non-blocking linked lists. In Distributed Computing, 15th International Conference, volume 2180 of Lecture ...
citeseer.ist.psu.edu/**harris**01**pragmatic**.html - 20k - Cached - Similar pages

l harris - ResearchIndex document query
A **pragmatic implementation** of non-blocking www.cl.cam.ac. uk/~tlh20/papers/**tim-harris**-caslists.ps.gz Integrating Task and Data Parallelism in UC - Dhagat, ...
citeseer.ist.psu.edu/cis?q=L.+Harris - 20k - Cached - Similar pages

Computer Laboratory - Practical lock-free data structures
A **Pragmatic Implementation** of Non-Blocking Linked Lists **Tim Harris** Proceedings of the 2001 IEEE Symposium on Distributed Computing ...
www.cl.cam.ac.uk/Research/SRG/netos/lock-free/ - 13k - Cached - Similar pages

Memory management
In preparation – contact **Tim Harris** for information ... A **Pragmatic Implementation** of Non-Blocking Linked Lists Timothy L **Harris** ...
www.cl.cam.ac.uk/Research/SRG/netos/misc/mm.html - 8k - Cached - Similar pages
[More results from www.cl.cam.ac.uk]

DBLP: Timothy L. **Harris**
2, EE, Timothy L. **Harris**: A **Pragmatic Implementation** of Non-blocking Linked-Lists. DISC 2001: 300-314. 2000. 1, Timothy L. **Harris**: Dynamic Adaptive ...
www.informatik.uni-trier.de/~ley/ db/indices/a-tree/h/Harris:Timothy_L=.html - 8k - Cached - Similar pages

SpringerLink - Chapter
A **Pragmatic Implementation** of Non-blocking Linked-Lists ... A1 University of Cambridge Computer Laboratory, Cambridge, UK, **tim.harris**@cl.cam.ac.uk ...
link.springer-ny.com/link/service/ series/0558/papers/2180/21800300.pdf - 21k - Cached - Similar pages

Program
11:35 - 11:45. BREAK (no coffee). 11:45 - 13:00. Session 9. 11:45. Timothy **Harris**. A **Pragmatic Implementation** of Non-Blocking Linked Lists ...
disc2001.di.fc.ul.pt/program.htm - 42k - Cached - Similar pages

$Id: netos.bib,v 1.7 2002/08/28 13:07:54 smh22 Exp ...
File Format: Unrecognized - View as HTML
... @InProceedings{**harris**:**pragmatic**, title = "A **Pragmatic Implementation** of ... author = "**Tim** D. Moreton and Ian A. Pratt and Timothy L. **Harris**", ...
ftp.fi.muni.cz/pub/bibliography/Os/NetOS.bib.gz - Similar pages

[PDF] A **Pragmatic Implementation** of Non-blocking Linked-Lists
File Format: PDF/Adobe Acrobat
Cambridge, UK, **tim.harris**@cl.cam.ac.uk. Abstract. We present a new non-blocking **implementation** ... A **Pragmatic Implementation** of Non-blocking Linked-Lists ...
www.springerlink.com/index/DG5QDE5TWEPRG3GM.pdf - Similar pages

Terms used **maged michael hash table**

Found **21,130** of **171,143**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Session 3: High performance dynamic lock-free hash tables and list-based sets

Maged M. Michael

August 2002 **Proceedings of the fourteenth annual ACM symposium on Parallel algorithms and architectures**

Publisher: ACM Press

Full text available: pdf(238.11 KB) Additional Information: full citation, abstract, references, citings, index terms

Lock-free (non-blocking) shared data structures promise more robust performance and reliability than conventional lock-based implementations. However, all prior lock-free algorithms for sets and hash tables suffer from serious drawbacks that prevent or limit their use in practice. These drawbacks include size inflexibility, dependence on atomic primitives not supported on any current processor architecture, and dependence on highly-inefficient or blocking memory management techniques.Building on ...

2 Split-ordered lists: lock-free extensible hash tables

Ori Shalev, Nir Shavit

July 2003 **Proceedings of the twenty-second annual symposium on Principles of distributed computing**

Publisher: ACM Press

Full text available: pdf(1.06 MB) Additional Information: full citation, abstract, references, citings, index terms

We present the first lock-free implementation of an extensible hash table running on current architectures. It provides concurrent insert, delete, and search operations with an expected $O(1)$ cost. It consists of very simple code, easily implementable using only load, store, and compare-and-swap operations. The new mathematical structure at the core of our algorithm is *recursive split-ordering*, a way of ordering elements in a linked list so that they can be repeatedly "split" using ...

Keywords: Compare-and-Swap, Concurrent Data Structures, Hash Table, Non-blocking Synchronization, Real-Time

3 Scalable lock-free dynamic memory allocation

Maged M. Michael

June 2004 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2004 conference on Programming language design and implementation PLDI '04**, Volume 39 Issue 6

Publisher: ACM Press

Full text available: pdf(213.94 KB) Additional Information: full citation, abstract, references, citings, index terms

Dynamic memory allocators (malloc/free) rely on mutual exclusion locks for protecting the consistency of their shared data structures under multithreading. The use of locking has many disadvantages with respect to performance, availability, robustness, and